UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

GRADO EN INGENIERÍA INFORMÁTICA

Amenazas y defensa en sistemas ciber-físicos con conexión al mundo virtual

Modelos de amenazas específicos para sistemas CPS

Threats and defense in cyber-physical systems with connection to the virtual world

Specific threat models for CPS systems

Realizado por
Pablo Gutiérrez Ruiz

Tutorizado por
María Cristina Alcaraz Tello
Francisco Javier López Muñoz

Departamento
Lenguajes y Ciencias de la Computación
UNIVERSIDAD DE MÁLAGA

MÁLAGA, Septiembre de 2020

Resumen

La aplicación de la Industria 4.0 a los hospitales conduce al término *smart hospitals*. Además, en una intervención quirúrgica se plantea un escenario crítico que es necesario monitorizar en todo momento. Por lo tanto, con el uso de sensores interconectados, se pueden obtener constantemente datos del entorno. Al estar monitorizando dicha información del entorno y estos datos estar comunicándose entre distintos subsistemas, la seguridad juega un papel importante en dichas comunicaciones. En este TFG (Trabajo de Fin de Grado), por tanto, se propone representar un quirófano mediante un escenario 3D, de modo que se puedan monitorizar y visualizar los datos contextuales que ocurren durante una intervención quirúrgica. Además de analizar qué puntos de las comunicaciones del sistema son susceptibles de ataques.

El escenario 3D se ha desarrollado mediante las herramientas de Blender y Unity, que han permitido no solo modelar el escenario, sino, interactuar con él. La información del entorno que se ha monitorizado se ha obtenido mediante sensores conectados a placas Arduino, las cuales han recogido y almacenado en la base de datos toda la información relativa al entorno.

La base de datos utilizada ha sido MongoDB, una base de datos no relacional la cuál permite almacenar de forma correcta la información proporcionada por los sensores. Esta ofrece una serie de soluciones referentes a la seguridad que han sido establecidas para dotar de seguridad al sistema.

Palabras clave: Sistemas Ciber-Físicos, MongoDB, Ataques, Modelado 3D

Abstract

The application of Industry 4.0 to hospitals leads to the term smart hospitals. Moreover, in a surgical intervention there is a critical scenario that must be monitored at all times. Hence, with interconnected sensors, it is possible to collect data from the environment. Due to environmental information is being monitored and this data is communicating between different subsystems, security plays an important role in said communications. Thus, in this FDP (Final Degree Project), it proposed to represent an operating room using a 3D scene in order that the contextual data that occurs during a surgical intervention can be monitored and visualized. In addition to this, system communications will be analysed in order to detect vulnerable points.

The 3D scene has been developed using Blender and Unity, which have allowed not only modeling the stage, but also interacting with it. The information of the environment has been obtained by sensors connected to Arduino boards, which have collected and stored all the information related to the environment in the database.

The database used has been MongoDB, NoSQL database which allows the information provided by the sensors to be stored correctly. It provides some security-related solutions that have been established to provide security to the system.

Key words: Cyber-Physical Systems, MongoDB, Attacks, 3D modelling

Índice

Índice de tablas

Índice de figuras

Lista de acrónimos

- **CPS**: Cyber-Physical System
- **DE**: Detectar
- **ENISA**: European Union Agency for Cybersecurity
- **FDG**: Final Degree Project
- **HIPAA**: Health Insurance Portability and Accountability Act
- **HMI**: Human-Machine Interface
- **ID**: Identificar
- **IP**: Internet Protocol
- **JSON**: JavaScript Object Notation
- **MitM**: Man in the Middle
- **NIST**: National Institute of Standards and Technology
- **NSF**: National Science Foundation
- **PR**: Proteger
- **RS**: Responder
- **SSL**: Secure Sockets Layer
- **SQL**: Structured Query Language
- **TCP**: Transmission Control Protocol
- **TFG**: Trabajo de Fin de Grado
- **TLS**: Transport Layer Security

1. Introducción

A continuación se va a mostrar la motivación que ha llevado a realizar este TFG, el estado del arte en el que se encuentra y los objetivos del mismo, así como la estructura de este documento.

1.1. Motivación

Hoy en día, nos encontramos ante la cuarta revolución industrial, la Industria 4.0, la cuál integra tecnologías inteligentes para poder mejorar los procesos clásicos ya existentes [12]. Este término está muy ligado a los sistemas CPS (Cyber-Physical Systems), el cual está definido por la NSF (National Science Foundation) como un sistema que integra algoritmos computaciones con componentes físicos [16] y una de las ventajas que ofrece es la seguridad de los sistemas. Esto llevó a investigar sobre cómo se podría adaptar estos sistemas al entorno de la salud, donde se encontró el término *smart hospitals*.

Se identificó un escenario crítico en el ámbito de la salud, el cual fue una sala de operaciones durante una intervención quirúrgica. Un entorno se define como crítico en el ámbito de la salud según la ENISA (European Union Agency for Cybersecurity) si una interrupción o un mal funcionamiento de un recurso no solo afecta al sistema si no también a los pacientes [2], y en una intervención quirúrgica cualquier cambio del entorno puede repercutir en el paciente.

Además, se contaba con la posibilidad de entrevistarnos con un cliente real (que por protección de datos no se va a nombrar), cirujano en el hospital de alta resolución de Benalmádena, lo cual hizo poder tener un acercamiento

de este TFG con el escenario real.

1.2. Estado del arte

Los hospitales, al igual que la industria, comienzan a estar altamente interconectados debido a los numerosos dispositivos que se comienzan a instalar. Estos dispositivos mejoran la calidad y la seguridad en los distintos ámbitos dentro del hospital, pero a su vez requieren de una seguridad extra para mantener dichos dispositivos. Por ejemplo, en el hospital Clínic de Barcelona se encuentras integrados a fecha de diciembre de 2019, 5000 elementos interconectados como servidores u ordenadores [28].

Además, en 2016, la ENISA publicó que el 67 % de las personas que entrevistó sobre hospitales inteligentes, declaró que los dispositivos conectados a la red eran recursos críticos [2].

El 22 de enero de 2020, el hospital de Torrejón fue el primer hospital español en ser secuestrado por un virus informático. Dicho ataque se cree que pudo haberse realizado intencionalmente desde dentro del hospital, que como se verá en el capítulo 6, siguen la modalidad de ataques *insider* y son más frecuentes de lo esperado. Con la actual pandemia mundial que se vive debido a la Covid-19, los hospitales están siendo el centro de los ataques, intentando así acceder a las redes de estos para poder controlar los datos [15].

Desde el punto de vista de la investigación, ya en 2012 se consideraba imprescindible tratar el tema de la seguridad en los sensores inalámbricos [19]. También se considera importante la inyección de datos falsos en los

sistemas de salud [5], tema tratado en este TFG.

Debido a esto se ve necesario el tener monitorizada una sala quirúrgica y controladas las comunicaciones entre los distintos dispositivos como son los sensores. Una amenaza a estos dispositivos puede desencadenar problemas mayores que puede conllevar la pérdida de vidas humanas.

1.3. Objetivos del TFG

El entorno crítico que conlleva una intervención quirúrgica hace necesaria la protección del mismo, y en esto se centran los objetivos de este TFG.

En primer lugar, se va a representar mediante una simulación 3D el quirófano en tiempo real. Para ello, el entorno estará sensorizado y toda está información será trasladada a la simulación para que pueda ser visualizada por el personal sanitario en el transcurso de una intervención quirúrgica.

En segundo lugar, se estudiarán las amenazas que se pueden llevar a cabo dentro de un entorno crítico como es una intervención quirúrgica. Además, se realizarán algunos ataques a la infraestructura que conlleva todo el sistema implementado.

Por último, el sistema implementado incluye una funcionalidad que es la detección de anomalías durante una intervención quirúrgica mediante el uso de *machine-learning*. Al tratarse de un trabajo en grupo, esta funcionalidad será llevada a cabo en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning.

Para resumir los objetivos del TFG se tendría:

- Representación 3D en tiempo real del quirófano (llevado a cabo en este TFG y en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning.).
- Ataque a la infraestructura, principal objetivo de este TFG (realizado en este TFG).
- Detección de anomalías mediante el uso de *machine-learning* (realizado en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning).

1.4. Estructuración de la memoria

Este TFG consta de siete capítulos donde se presenta todo lo relacionado con este trabajo. Las secciones son las siguientes:

- Capítulo 1 - Introducción: se presenta la motivación de este TFG junto con sus objetivos. Además de esto, el capítulo aborda el estado del arte en lo referente a los *smart hospitals*.
- Capítulo 2 - Metodología: En este capítulo se aborda la explicación de la metodología usada y como se ha llevado a cabo.
- Capítulo 3 - Digitalización del mundo real: Se encuentran los requisitos del sistema y el caso de uso.
- Capítulo 4 - Tecnologías usadas: Se exponen las tecnologías hardware, software y de base de datos.
- Capítulo 5 - Arquitectura del sistema: Se muestra la arquitectura de las placas Arduino y sensores, la estructuración de la base de datos

no relacional, tanto su creación como diseño y comunicaciones, y el escenario 3D.

- Capítulo 6 - Mecanismos de ataque: Tipos de amenazas que existen en un sistema CPS como el expuesto en este TFG, así como la seguridad que debe existir en las comunicaciones.

- Capítulo 7 - Conclusiones y trabajos futuros: Para finalizar el TFG se encuentran las conclusiones del mismo y como podría ser mejorado y completado en futuros trabajos.

2. Metodología

En este proyecto se ha aplicado una metodología ágil basada en Scrum [23]. Esta, se ha aplicado en iteraciones de dos semanas en las cuales nos reuníamos los tutores y los alumnos que estábamos realizando el trabajo.

Los tutores han desarrollado el rol de Scrum Master supervisando el trabajo realizado por los alumnos. Estos últimos han desarrollado el rol del equipo de desarrollo.

Dado que este trabajo es llevado a cabo en equipo, la organización entre todos los integrantes es fundamental y por ello, se ha utilizado la herramienta Trello [7]. Trello nos ofrece un tablero de Kanban [8] (figura 1), el cual permite organizar las distintas tareas en columnas. Estas columnas pueden ser 'Lista de tareas', 'En proceso', 'Hecho' o cualquier otra columna que favorezca el trabajo. Todos los integrantes tienen acceso a este tablero, lo cual permite al equipo de desarrollo tener una visión global de todas las tareas del proyecto y a los Scrum Master supervisar como va evolucionando el trabajo durante los sprints.

Figura 1: Ejemplo de tablero Kanban en Trello

Durante las dos semanas que duraba cada sprint, había reuniones intermedias con los Scrum Master para resolver problemas que iban surgiendo y añadir tareas que estarían presentes en próximos sprints. Al finalizar los sprints, las tareas no finalizadas se incorporaban al siguiente sprint.

La planificación temporal de todas las fases se muestra en el diagrama de Gantt de la figura 2.

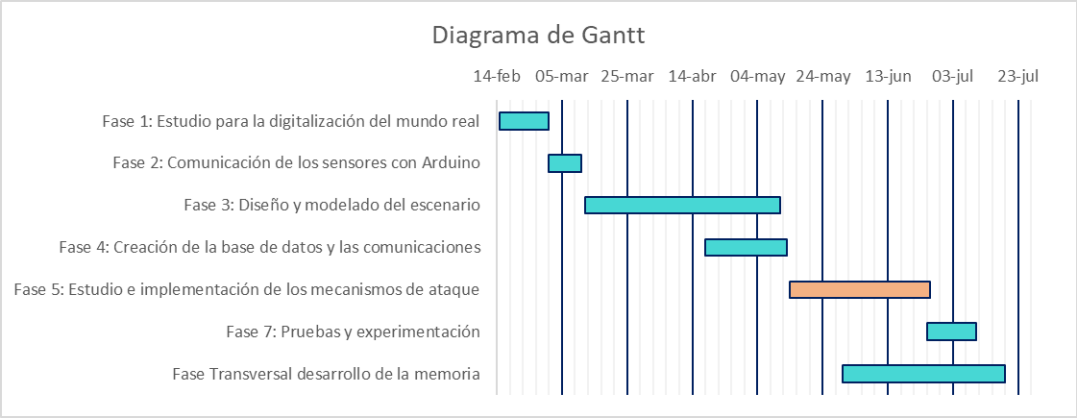


Figura 2: Diagrama de Gantt

3. Digitalización del mundo real

En este capítulo se van a presentar los requisitos, cómo han sido obtenidos y los casos de uso.

3.1. Requisitos

Los requisitos de este TFG han sido obtenidos del Marco para la mejora de la seguridad cibernética en infraestructuras críticas del NIST (National Institute of Standards and Technology) [3].

Además, en cada requisito se ha mostrado la regla HIPAA (Health Insurance Portability and Accountability Act) [1] a la que se le asocia, siempre y cuando existiera una regla HIPAA para asociar. Estas reglas definen lo que tiene que cumplir un sistema de salud y para estos requisitos concretamente se han usado las reglas del apartado Seguridad y Privacidad de la refenrecia [1].

Estos requisitos están organizados en cuatro tablas:

- Identificar ID (tabla 1): son aquellos requisitos referentes a comprender el entorno.
- Proteger PR (tabla 2): tratan de contener los posibles ataques al sistema.
- Detectar DE (tabla 3): son aquellos que tratan el monitoreo y la detección de anomalías.

- Responder RS (tabla 4): tratan de comunicar las anomalías que ocurren en el sistema.

GESTIÓN DE ACTIVOS			
ID.AM-6	Estarán descritos todos los roles que actúan en el sistema.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información) Regla HIPAA - § 164.308 Administrative safeguards. (a.3.i)		
Afecta a	Sistema CPS (Base de datos)	Administrador	Personal Sanitario
Precondición	Hay distintos usuarios, dispositivos y procesos que deben tener asignados permisos y privilegios adecuados.		
Postcondición	Los usuarios deberán tener asignados los roles correspondientes.		

ENTORNO EMPRESARIAL			
ID.BE-2	Se establecerá el escenario en el que se sitúa el sistema desarrollado y los parámetros que se monitorizarán.		
Afecta a	Sistema CPS		
Precondición	El objetivo del sistema será establecido		
Postcondición	Quedará establecido el entorno de actuación del sistema, en este caso una sala quirúrgica.		

EVALUACIÓN DE RIESGOS			
ID.RA-3	Las vulnerabilidades a las que Sistema CPS tiene que hacer frente serán identificadas.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.2)*(Protección de la integridad de los datos)		
Afecta a	Sistema CPS		
Precondición	Al tratarse de una infraestructura crítica habrá que analizar todos los posibles riesgos y amenazas a los que Sistema CPS puede estar expuesto		
Postcondición	Las medidas para prevenir las vulnerabilidades serán implementadas.		

Tabla 1: Requisitos de identificación

Seguridad de los datos (PR.DS)			
PR.AC-1	Los usuarios y contraseñas se gestionarán para dispositivos, usuarios y procesos autorizados (estos usuarios y contraseñas estarán gestionados por MongoDB).		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3)(Protección ante usos y revelación de información) Regla HIPAA - § 164.308 Administrative safeguards. (a.3.i)		
Afecta a	Sistema CPS	Personal sanitario y técnico	
Precondición	Identificar qué dispositivos, usuarios y procesos tienen que tener acceso a la base de datos.		

Postcondición	Cada dispositivo, usuario y proceso tendrá un usuario y contraseña.		
PR.AC-2	En el momento en el que el sistema esté implementado en un hospital real, el acceso a los dispositivos y sensores deberá estar restringido.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información)		
Afecta a	Hospital		
Precondición	Establecer las medidas de restricción de acceso a dispositivos y sensores.		
Postcondición	Los dispositivos y sensores contarán con las medidas de protección adecuadas para acceder a ellos y la información que contienen.		
PR.AC-4	Los accesos a la base de datos garantizarán el principio de mínimos privilegios.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información) Regla HIPAA - § 164.308 Administrative safeguards. (a.3.i)		
Afecta a	Sistema CPS (Base de datos)	Personal sanitario y técnico	
Precondición	Establecer los usos de la base de datos de cada usuario.		

Postcondición	Los usuarios contarán con aquellos privilegios que le permitan acceder exclusivamente a aquellos recursos a los que está autorizado.		
PR.AC-6	Sistema CPS verificará todas las credenciales de seguridad introducidas en el sistema.		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.3) (Protección ante usos y revelación de información)		
Afecta a	Sistema CPS	Personal sanitario y técnico	
Precondición	Un usuario de Sistema CPS ha sido dado de alta, tendrá un nombre de usuario y una contraseña.		
Postcondición	El usuario está autenticado		
SEGURIDAD DE LOS DATOS			
PR.DS-2	Las comunicaciones de los datos estarán protegidas		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.1) (a.2) (Protección de la integridad de los datos)		
Afecta a	Sistema CPS		
Precondición	Se habrá obtenido un certificado SSL para cifrar las comunicaciones de datos de los sensores hasta la base de datos.		
Postcondición	La comunicación de los datos queda protegida..		

Tabla 2: Requisitos de protección

ANOMALÍAS Y EVENTOS			
DE.AE-3	Los datos de sensores y de los paquetes TCP/IP entre los sensores y la base de datos serán recopilados y almacenados en la base de datos.		
Afecta a	Sistema CPS		
Precondición	Existe un flujo de datos de sensores y una comunicación con la base de datos a través de la red.		
Postcondición	La información queda almacenada de forma estructurada para su posterior utilización.		
DE.AE-5	Se analizarán posibles anomalías y establecerán umbrales de alerta mediante algoritmos de machine learning.		
Afecta a	Sistema CPS		
Precondición	Han sido establecidas las condiciones normales del escenario y los algoritmos de machine learning han sido entrenados.		
Postcondición	Las anomalías deberán ser detectadas correctamente.		
MONITOREO CONTINUO DE LA SEGURIDAD			
DE.CM-1	La red será monitorizada para detectar posibles ataques		
Se asocia a	Regla HIPAA - § 164.306 Security standards: General rules. (a.2) (a.3) (Protección de la integridad y, el uso y revelación de información)		
Afecta a	Sistema CPS		
Precondición	Se producen comunicaciones a través de la red.		
Postcondición	El tráfico es analizado para detectar anomalías.		
DE.CM-2	La sala quirúrgica será monitorizada mediante una representación adecuada		

Afecta a	Sistema CPS	Sala quirúrgica	
Precondición	Los sensores recogen datos del entorno físico.		
Postcondición	Se pueden observar los cambios en el entorno físico en tiempo real.		
PROCESOS DE DETECCIÓN			
DE.DP-3	Los procesos de detección serán probados.		
Afecta a	Sistema CPS		
Precondición	Se establecen los procesos necesarios de detección.		
Postcondición	Los procesos están verificados para su utilización.		
DE.DP-4	Se especificará el origen de la anomalía.		
Se asocia a	Regla HIPAA - § 164.404 Notification to individuals		
Afecta a	Sistema CPS	Personal sanitario	
Precondición	Se ha producido una anomalía		
Postcondición	Se conoce de qué elemento de Sistema CPS procede la situación anómala.		
DE.DP-5	Los algoritmos de detección serán mejorados periódicamente.		
Afecta a	Sistema CPS		
Precondición	Se establece la frecuencia de mejora de los algoritmos de detección.		
Postcondición	Los algoritmos quedan actualizados.		

Tabla 3: Requisitos de detección

COMUNICACIONES			
RS.CO-1	Los responsables deberán conocer sus roles a la hora de tomar una decisión ante una incidencia.		
Se asocia a	Regla HIPAA - § 164.308 Administrative safeguards. (a.5. i)		
Afecta a	Sistema CPS	Personal sanitario	
Precondición	Se estudian las responsabilidades de cada usuario		
Postcondición	El sistema es utilizado correctamente.		
RS.CO-2	Las anomalías serán notificadas mediante la pantalla de la simulación.		
Afecta a	Sistema CPS	Personal sanitario	
Precondición	Se detecta una anomalía.		
Postcondición	El personal queda informado de la anomalía y actuará en consecuencia.		

Tabla 4: Requisitos de respuesta

3.2. Caso de uso

Este TFG y el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning, asientan las bases de un sistema monitorización del entorno de una sala quirúrgica. Este sistema tiene las funcionalidades que se ven en el caso de uso de la figura 3 y los actores que afectan al sistema son los siguientes:

- **Personal sanitario:** grupo de personas que tienen acceso al ordenador

situado en la sala de operaciones.

- **Arduino:** Placas de Arduino (Uno y Mega) que recogen los datos de los sensores.
- **Sniffer:** Representa el código de Python que recoge los datos de red.
- **Administrador/Técnico:** Personal que tiene la función de administrar los algoritmos de detección de anomalías.
- **Base de datos:** Representa el lugar físico donde se almacenan los datos de sensores, red y predicciones.

El caso de uso que se muestra en la figura 3 hace referencia al sistema de monitorización. Desde el punto de vista del personal sanitario es posible explorar el escenario (sala quirúrgica) como se muestra en la sección 5.3 y comenzar el proceso de detectar anomalías mientras se encuentran en una intervención quirúrgica.

Por otro lado, los actores de Arduino y Sniffer intervienen en el buen funcionamiento de la detección de anomalías recogiendo y enviando los datos del entorno y red.

Por ultimo, el actor Administrador/Técnico será el encargado de gestionar los modelos de predicción. Como se ha comentado anteriormente, la detección de anomlías se realiza en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning.

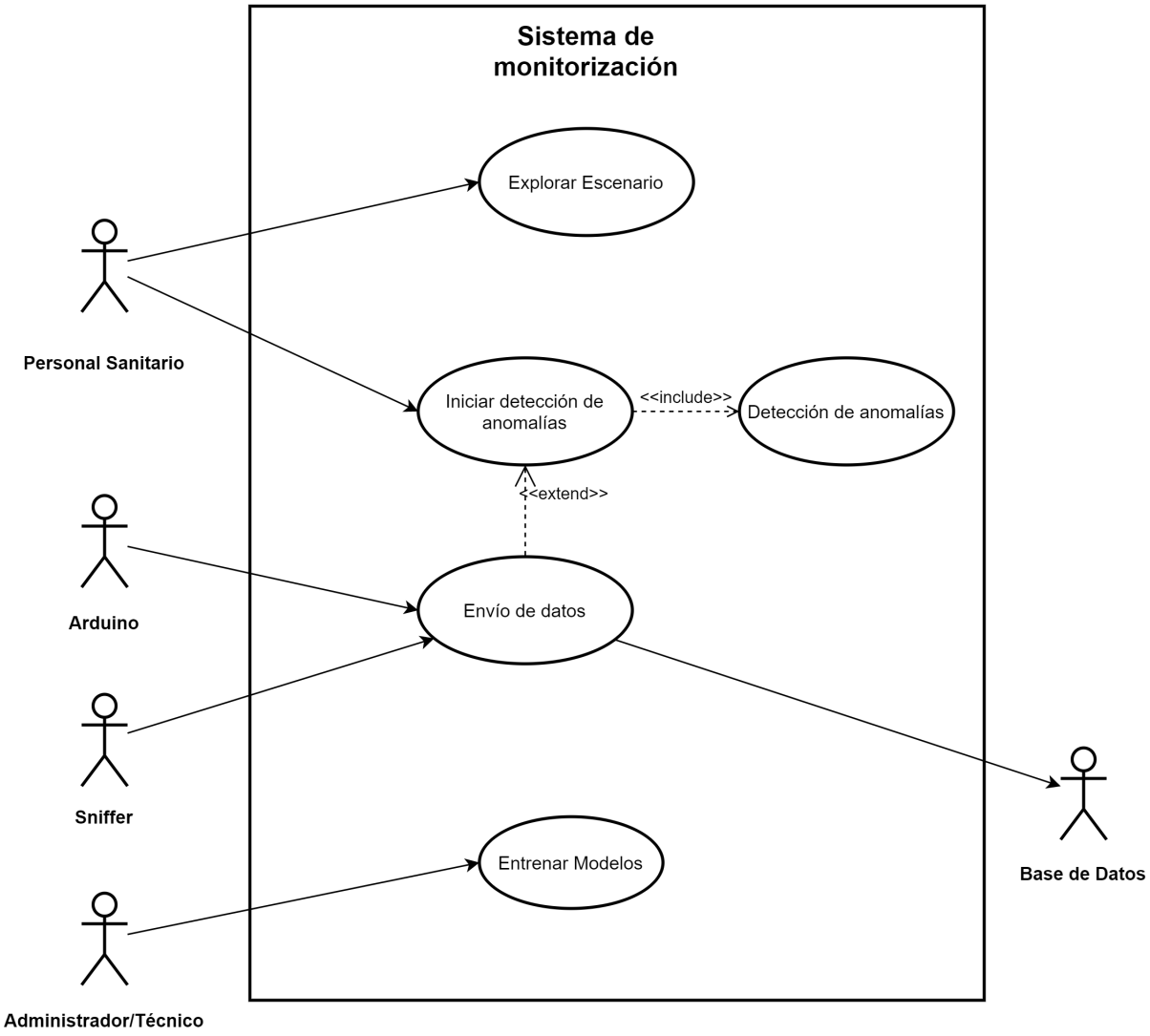


Figura 3: Caso de uso

4. Tecnologías hardware y software

En esta sección se van presentar las diferentes tecnologías tanto hardware como software que han sido utilizadas para la realización de este TFG.

4.1. Componentes hardware

Al tener que representar una situación del mundo real mediante un escenario virtual 3D, ha sido necesario el uso de diferente hardware que permitiera obtener datos del entorno como son distancias, temperatura y humedad. El hardware que ha permitido realizar esta tarea son dos placas Arduino y una serie de sensores que se exponen a continuación.

4.1.1. Arduino

Figura 4: Arduino Mega

Arduino es una placa programable que nos permite en este caso, interactuar con los distintos sensores que captan la información del entorno.

Ha sido necesario el uso de dos placas Arduino, una Arduino Uno y otra

Arduino Mega. En un principio se contaba con solamente una placa (Arduino Uno), pero la limitación de pines digitales, SDA y SCL hizo necesario adquirir una segunda placa Arduino Mega.

- Arduino Uno

  Esta placa es con la que se trabajó inicialmente, pero hubo la necesidad de incorporar más pines. Las principales características son que incorpora 16 pines entrada/salida digitales, 1 pin SDA, 1 pin SCL y ofrece un voltaje de 5V.

- Arduino Mega

  Esta placa fue por la que se optó a la hora de ampliar el número de pines. Igual que la placa Arduino Uno nos ofrece un voltaje de 5V, 1 pin SDA y 1 pin SCL, pero esta vez cuenta con 54 pines digitales.

Los sensores usados para recoger la información del entorno han sido seis sensores de ultrasonido, dos sensores de temperatura y dos de humedad.

4.1.2. HC-SR04

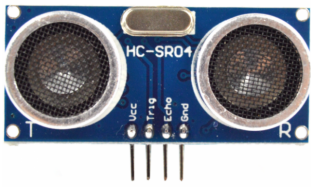


Figura 5: Sensor de ultrasonido HC-SR04

Con este sensor se ha monitorizado el movimiento de la mesa de operaciones. El sensor de ultrasonido emite una onda de ultrasonido, esta refleja

en algún objeto y el sensor la recibe. De esta forma, según la variación entre la emisión y la recepción de la onda, se puede hallar la distancia a la que se encuentra el objeto.

Este sensor es capaz de detectar objetos en un ángulo de 15º y en un rango de 2cm a 400cm [14].

4.1.3. Grove - Temperatura y humedad

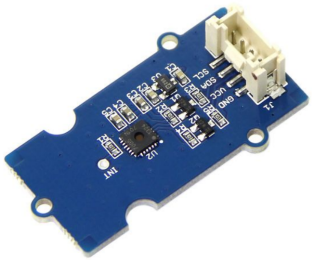


Figura 6: Sensor Grove - Temperatura y Humedad

Este sensor ha sido utilizado para obtener las medidas relativas a la temperatura y la humedad. Permite medir la temperatura en un rango de 0ºC a 70ºC y la humedad en un rango de 0 a 100 % [33].

Como característica principal, este sensor cuenta con una comunicación con Arduino mediante el bus I2C [20]. La conexión a la placa de Arduino se realiza mediante dos únicos cables. Estos cables son:

- SCL: Señal de reloj.
- SDA: Envío de datos.

Las placas Arduino utilizadas, como se menciona en la sección 4.1.1, solo cuentan con una entrada SCL y SDA. Cuándo esto ocurre, si se quieren conectar varios sensores a un mismo bus, es necesario que cada sensor tenga una dirección única. Los sensores utilizados resultaron tener la misma dirección física, así que se optó por incluir la segunda placa Arduino para solventar dicho problema.

Aunque se adquirieron tres sensores de temperatura y humedad, en el sistema solo se pudieron incluir dos debido a la imposibilidad de adquirir una tercera placa Arduino.

4.2. Lenguajes de programación

4.2.1. Arduino

Arduino es un lenguaje que deriva y comparte sintaxis con C++, y que nos permite interactuar con las distintas placas Arduino [6], en este TFG, Uno y Mega. Como característica principal, a la hora de programar se definen dos métodos: setUp() y loop().

- setUp(): se ejecuta la primera vez que inicia la placa y suele ser utilizado para definir pines de entrada y/o salida.
- loop(): una vez finalizado el método setUp(), loop() se ejecuta en bucle hasta que la placa no se apague. En este TFG, en el método loop() se está constantemente leyendo datos de los sensores.

4.2.2. Python

Python es un lenguaje multiparadigma [30] que tiene un gran número de librerías para tratar diversos temas específicos. Concretamente en este TFG, su facilidad para trabajar con diccionarios y objetos JSON (JavaScript Object Notation) han sido muy adecuados a la hora de manejar los objetos de la base de datos, además, su librería pymongo facilita el hecho de insertar y leer de la base de datos. Siguiendo por esta misma línea, con la librería Scapy [31], trabajar en Python con los objetos de los paquetes de red ha sido muy cómodo por el hecho de poder acceder a ellos como si de un diccionario se tratasen.

4.2.3. C#

C# es un lenguaje multiparidigma [32] y del que hace uso Unity [34] para comunicarse con el motor gráfico [35]. Para establecer comunicación con la base de datos MongoDB [25] ha sido necesario instalar los drivers de MongoDB para C#.

4.3. Librerías

4.3.1. PyMongo

PyMongo [29] es una librería de Python la cual permite trabajar con MongoDB desde Python. Esta ha sido usada para establecer conexión con la base de datos, escribir y leer datos de ella.

4.3.2. Scapy

Scapy [31] es una librería de Python que nos permite, entre otras utilidades, capturar tráfico de la red y enviar paquetes.

Esta librería incluye numerosos protocolos, aunque no el usado principalmente en este TFG: MongoDB Wire Protocol [27].

4.4. MongoDB

MongoDB [25] ha sido la base de datos seleccionada para este proyecto. Se trata de una base de datos no relacional que trabaja con documentos (objetos) en formato JSON, lo cual la hace una muy buena opción para nuestro sistema, el cual va almacenar un documento con las medidas del entorno en cada instante de tiempo.

En lugar de tablas como en SQL (Structured Query Language), en MongoBD se cuenta con colecciones, que es donde se guardan los documentos JSON. Esto permite después, trabajar de forma cómoda tanto en Python como en C#.

4.5. Software

4.5.1. Wireshark

Wireshark [37] es uno de los analizadores de tráfico más usados. Para este proyecto presenta una gran ventaja y es que incluye el protocolo MongoDB Wire Protocol. Gracias a esto se pueden detectar muy fácilmente las trazas

de dicho protocolo. Además, resulta de gran utilidad para comprobar si las comunicaciones están o no cifradas, o si algún dato ha cambiado desde su envío.

4.5.2. MongoDB Compass

MongoDB Compass [26] es una herramienta gráfica que permite visualizar de manera amigable las distintas bases datos y colecciones de MongoDB. Además, permite realizar consultas en las distintas colecciones, insertar, eliminar y actualizar datos.

Para el diseño del escenario 3D se ha hecho uso de las herramientas Blender [9] y Unity.

4.5.3. Blender

Blender [9] es un software que permite el modelado 3D y entre sus principales características se encuentran en que es software libre y que los modelos creados pueden ser exportados a Unity.

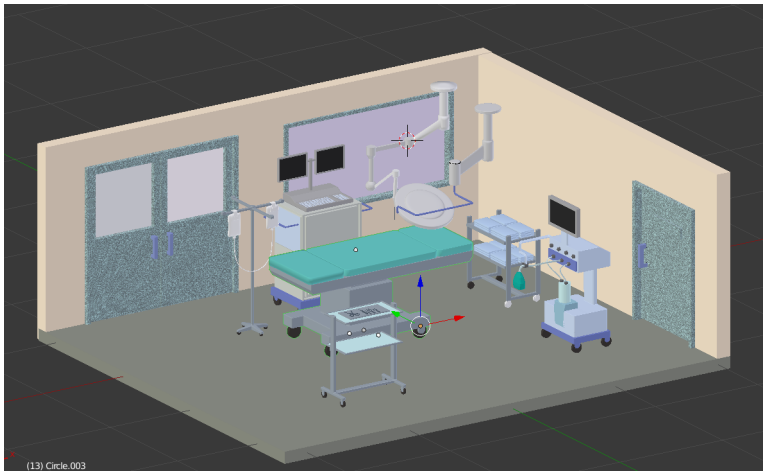


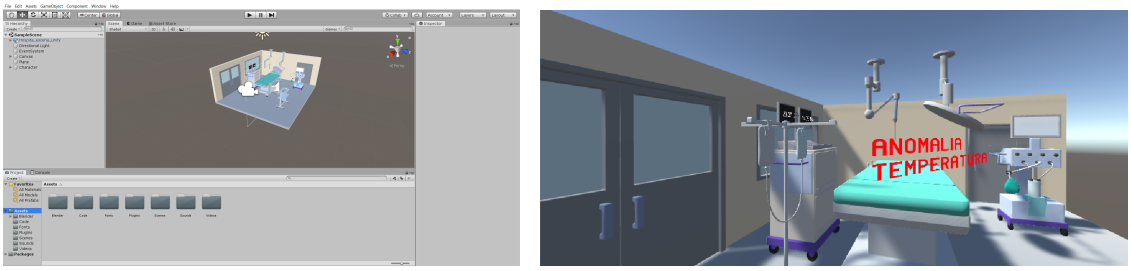
Figura 7: Escenario general en Blender

La representación, aunque fiel a la realidad, se ha tratado de realizar de forma sencilla para que no suponga una distracción al personal sanitario durante una intervención quirúrgica, como se puede ver en la figura 7, basada en [21].

4.5.4. Unity

Unity [34] es un motor gráfico el cuál ha permitido representar la información del mundo real en el escenario modelado en Blender. La principal ventaja que ofrece a este proyecto, es la capacidad de poder actualizar los valores de posiciones y texto en tiempo real, pudiendo así dar información adecuada al personal sanitario.

En Unity se ha representado el movimiento de la mesa de operaciones y los cambios en la temperatura y humedad, así como alertas sobre las anomalías que pueden ir ocurriendo en los sensores a lo largo de una intervención quirúrgica.



(a) Visión general de Unity

(b) Escena de Unity

Figura 8: Escenario Unity

5. Arquitectura del sistema

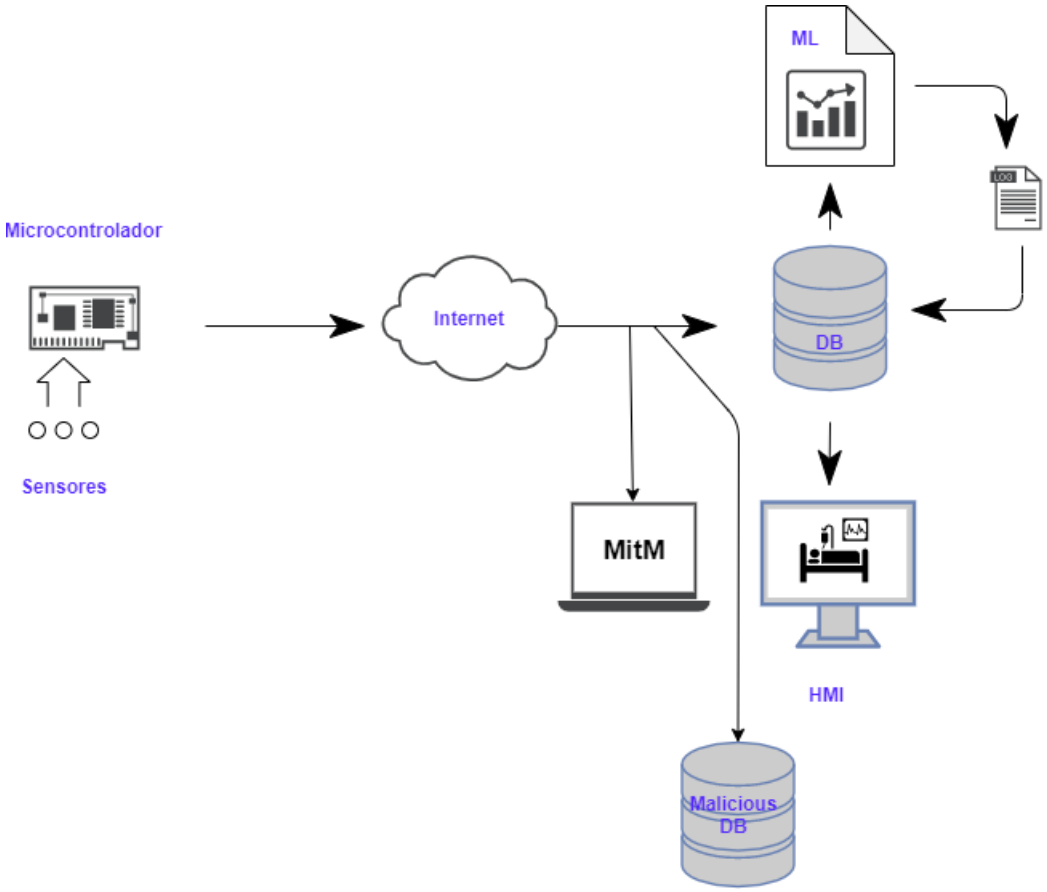


Figura 9: Esquema simplificado del sistema

La figura 9 muestra un esquema simplificado del sistema. De izquierda a derecha en la figura se encuentra el microcontrolador formado por dos placas Arduino (Uno y Mega) que son las encargadas de tratar los datos que reciben los sensores y enviarlos a través de internet a la base de datos (MongoDB). A su vez, la base de datos está comunicada con los algoritmos de machine-learning para proporcionarles los datos de los sensores y almacenar el resultado de las predicciones. El HMI (Human-Machine Interface), también

accede la base de datos para poder actualizar la información de posición de la mesa de operaciones, temperatura y humedad en el escenario 3D (Unity).

En la parte inferior de la figura se muestra un Man in the Middle (MitM) y una base de datos maliciosa (Malicious DB), esto se encuentra desarrollado en la sección 6.

El apartado de machine-learning que se encuentra en la parte superior de la imagen es explicada en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning.

5.1. Arduino y sensores

Como se ha hablado en la sección 4, las placas de Arduino usadas han sido la Uno y la Mega, y los sensores: Ultrasonido HC-SR04 y Grove - Temperatura y Humedad.

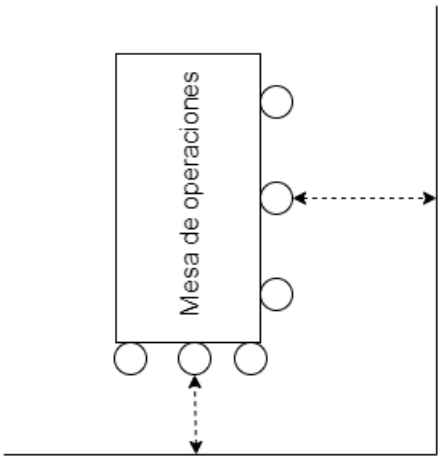


Figura 10: Representación de los sensores de ultrasonido HC-SR04

Los seis sensores de ultrasonido se sitúan como se indican en la figura 10,

de modo que un grupo de tres obtiene la distancia con respecto a un lado de la sala de operaciones y el otro grupo con respecto al otro lado. De este modo, la mesa de operaciones se puede representar sobre el plano en el que se encuentra el suelo.

Los dos sensores de temperatura y humedad, se distribuyen alejados en la sala de operaciones, de modo que se cubra el máximo espacio.

Los esquemas de los circuitos de ambas placas para tener todos los sensores conectados de forma correcta se pueden observar en las figuras 16 y 17 del manual de instalación. A través de la comunicación serie, Arduino se comunica con Python gracias a la librería Serial, para que este último mediante la librería PyMongo, inserte los datos de los sensores en la base de datos.

5.2. Base de datos no relacional

Como se ha mencionado en la sección 4.4, la base de datos elegida para este proyecto ha sido MongoDB.

5.2.1. Creación de la base de datos

Idealmente, la base de datos debería estar creada en una máquina perteneciente a la misma red donde se encontrase todo el modelo. Debido a la situación excepcional provocada por la COVID-19 y al tratarse de un TFG grupal en el que ambos integrantes debían poder conectarse a la base de datos, esta a necesitado ser accesible desde el exterior.

MongoDB se ha creado en un sistema Windows. Ha sido necesario esta-

blecer un regla TCP (Transmission Control Protocol, Protocolo de Control de Transmisión) en el firewall de Windows, a través de su interfaz gráfica, para que MongoDB pudiera recibir conexiones del exterior a través del puerto 27017. Además, en el router se ha establecido otra regla para que redirija las conexiones entrantes por el puerto 27017 a la IP del equipo donde se instalado MongoDB.

5.2.2. Diseño de la base de datos

La base de datos, llamada cps, cuenta con las siguientes colecciones:

- **packet_data_train**: colección donde se almacenan datos de los paquetes de red en condiciones normales usados posteriormente para detectar anomalías.
- **sensors_data_train**: colección donde se almacenan datos recibidos de los sensores en condiciones normales usados posteriormente para detectar anomalías.
- **packet_data**: colección donde se almacenan datos de los paquetes durante la intervención quirúrgica.
- **sensors_data**: colección donde se almacenan datos recibidos de los sensores durante la intervención quirúrgica.
- **predict_log**: colección donde se almacenan los resultados de las predicciones para poder ser representadas en la simulación de Unity.

Estas colecciones son accedidas desde el TFG de Marta Ferrer Cuesta,

Detección de anomalías en sistemas CPS mediante machine learning, y su estructura es la siguiente.

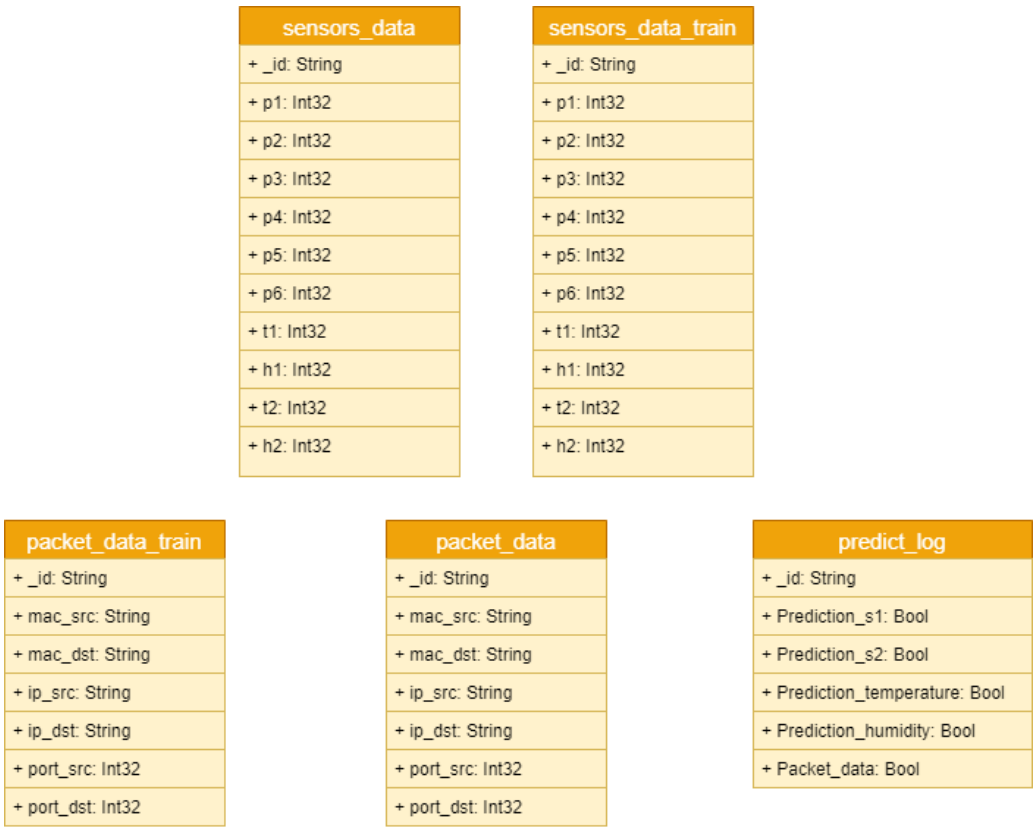


Figura 11: Colecciones de la base de datos

Para un ataque que será comentado posteriormente en la sección 6, se ha creado una base de datos "maliciosa" con la siguiente colección:

- **traffic_sniff**: colección que almacena una réplica de los datos introducidos en las colecciones **packet_data_train** y **sensors_data_train**.

Además de estas colecciones, se cuentan con los siguientes usuarios:

- **admin**: usuario con todos los privilegios encargado de gestionar la base de datos.
- **arduino**: usuario encargado de insertar los datos procedentes de los sensores en la base de datos. Tiene privilegios de lectura y escritura.
- **machine-learning**: usuario encargado de realizar el entrenamiento y las predicciones de las anomalías. Tiene privilegios de lectura y escritura.
- **unity**: usuario encargado de comunicarse con la base de datos para actualizar los datos en el escenario 3D y las alertas de anomalías. Cuenta con privilegios de lectura.
- **sniff**: usuario encargado de insertar los datos de red en la base de datos. Tiene privilegios de lectura y escritura.
- **mallory**: usuario que se comunica con la base de datos "maliciosa" mencionada anteriormente para replicar la base de datos original.

Estos usuarios, como se hablará en la sección 6.3, garantizan el principio de mínimos privilegios [4].

Usuario	Base de datos de acceso	Privilegios
admin	admin	“userAdminAnyDatabase” “readWriteAnyDatabase”
arduino	cps	“readWrite” : “cps”
machine-learning	cps	“readWrite” : cps
unity	cps	“read” : “cps”
sniff	cps	“readWrite”: “cps”

Tabla 5: Usuarios de la base de datos

5.2.3. Comunicaciones con las bases de datos

La base de datos se comunica con los siguientes subsistemas:

- Datos de los sensores.
- Datos de red.
- Machine - learning.
- Escenario 3D.

Datos de los sensores: Las placas Arduino usadas no cuentan con conexión a Internet, por tanto el envío de los datos de los sensores se ha realizado a través de un script de Python el cuál se comunica con la base de datos y con las placas Arduino. Este script hace uso de las librerías serial para comunicarse con las placas Arduino y PyMongo para insertar los dados en la base de datos

Datos de red: Los datos de red son guardados en la base de datos mediante un script de Python, el cual hace uso de las librerías Scapy y PyMongo.

Machine-learning: Los datos que se necesitan para realizar las detecciones de anomalías se obtienen mediante un script de Python que se conecta a la base de datos haciendo uso de la librería PyMongo.

Escenario 3D: El escenario 3D actualiza su información desde C# haciendo uso de los drivers de MongoDB para dicho lenguaje.

5.3. Creación del escenario 3D

El escenario creado ha sido basado en la imagen incluida como parte del trabajo [22]. A la hora de diseñar el quirófano se ha trabajado sobre un escenario sencillo sin excesivos detalles que hicieran al personal presente durante la operación tener demasiadas distracciones. Pero aún así, cuenta con los elementos esenciales que debe tener un quirófano conocidos tras la entrevista con el cliente (cirujano de hospital),

- Mesa de operaciones.
- Mesa de instrumentación.
- Portasuero.
- Lampara quirúrgica.
- Torre de gases.
- Monitor para monitorizar constantes vitales.

- Mueble con gasas.
- Carrito auxiliar con pantallas de información.
- Doble puerta de acceso al quirófano.
- Puerta de sucio (puerta que conduce a un pasillo de sucio por donde salen todos los utensilios utilizados durante una intervención quirúrgica).

Además de los elementos encontrados en la imagen [22], se ha añadido un brazo robot el cual podrá ser utilizado en alguno de los trabajos futuros mencionados en la sección 7.3.

En Blender se ha realizado únicamente la tarea de modelización del escenario. Después, ha sido exportado para poder ser utilizado en Unity y aportarle la funcionalidad del sistema.

Una vez importado en Unity, lo primero que se ha hecho ha sido establecer las texturas del escenario, siempre de manera no excesiva para no interferir en la atención del personal que se encuentra en la intervención quirúrgica (como se comenta unas líneas más arriba).

Lo siguiente que se ha realizado es representar la información de los sensores en el escenario 3D. Esta información es actualizada en cada frame de la simulación. Esto se realiza gracias a que en Unity los scripts derivan de la clase MonoBehaviour [36] la cual incluye los métodos Start() y Update(). La función Start() se ejecuta solamente una vez en el primer frame y la función Update() se ejecuta en cada frame, que es donde se actualiza la información de los sensores. Los datos de los sensores actualizados se visualizan de la siguiente manera:

- La información de los sensores de ultrasonidos es representada en la posición de la mesa de operaciones.
- La información de los sensores de temperatura y humedad son representados en las dos pantallas del escenario 3D.

En el momento que se produce una anomalía se muestra un texto flotante de color rojo indicando el texto “ANOMALÍA” junto con el sensor del que procede la anomalía, ejemplo: “ANOMALÍA HUMEDAD”. Este texto flotante va acompañado una alarma sonora por si el personal presente en la intervención quirúrgica no estuvieran atentos a la simulación en ese momento. Este texto se puede ver en la figura 8b.

Para mejorar la experiencia de la simulación, es posible navegar a través de la sala quirúrgica con las teclas W, A, S, D, y rotar la vista de la cámara con el movimiento del ratón.

A las opciones de detectar anomalías y navegar a través de la sala quirúrgica se acceden a través de un menú previo.

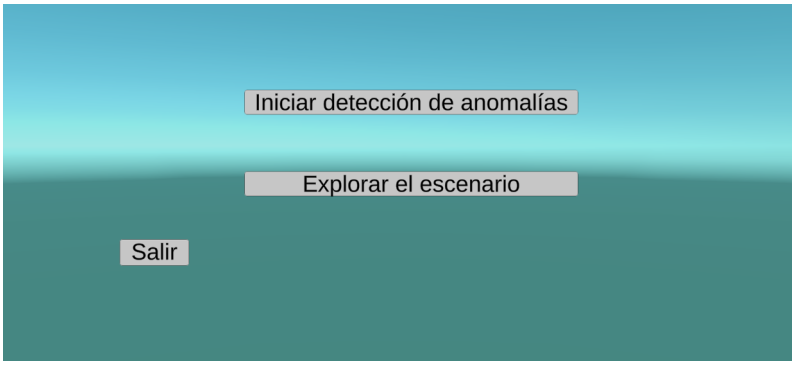


Figura 12: Menú del sistema

Iniciar detección de anomalías: Desde esta opción accedemos una

a escena de la sala quirúrgica que no es explorable, es decir, no podemos navegar a través de ella, pero se puede visualizar el movimiento de la mesa de operaciones y de los valores de temperatura y humedad. En esta escena, si ocurre una anomalía se muestra la alerta en pantalla.

El motivo por cual esta escena no es explorable, es para evitar posibles accidentes durante la intervención quirúrgica, pudiendo dejar a la simulación sin visión de los valores en tiempo real (cámara dada la vuelta).

Explorar el escenario: Esta opción permite acceder a la misma escena del caso anterior (iniciar detección de anomalías), pero en este caso si es explorable. Como se ha comentado anteriormente, esta navegación se realiza a través de las teclas W, A, S y D, y el movimiento del ratón.

Por último, se ha generado un fichero ejecutable para poder iniciar la simulación independientemente de disponer del entorno Unity.

6. Mecanismos de ataque

En esta sección se van a mostrar algunos de los riesgos con los que cuenta el sistema tratado en este TFG. También se van a identificar los posibles atacantes que podrían actuar en este sistema. Por último, se van a tratar las soluciones que ofrece MongoDB para protegerse contra distintos ataques.

6.1. Modelo de amenazas

Antes de detallar las amenazas concretas a las que el sistema CPS está expuesto, se van a localizar los puntos susceptibles del sistema.

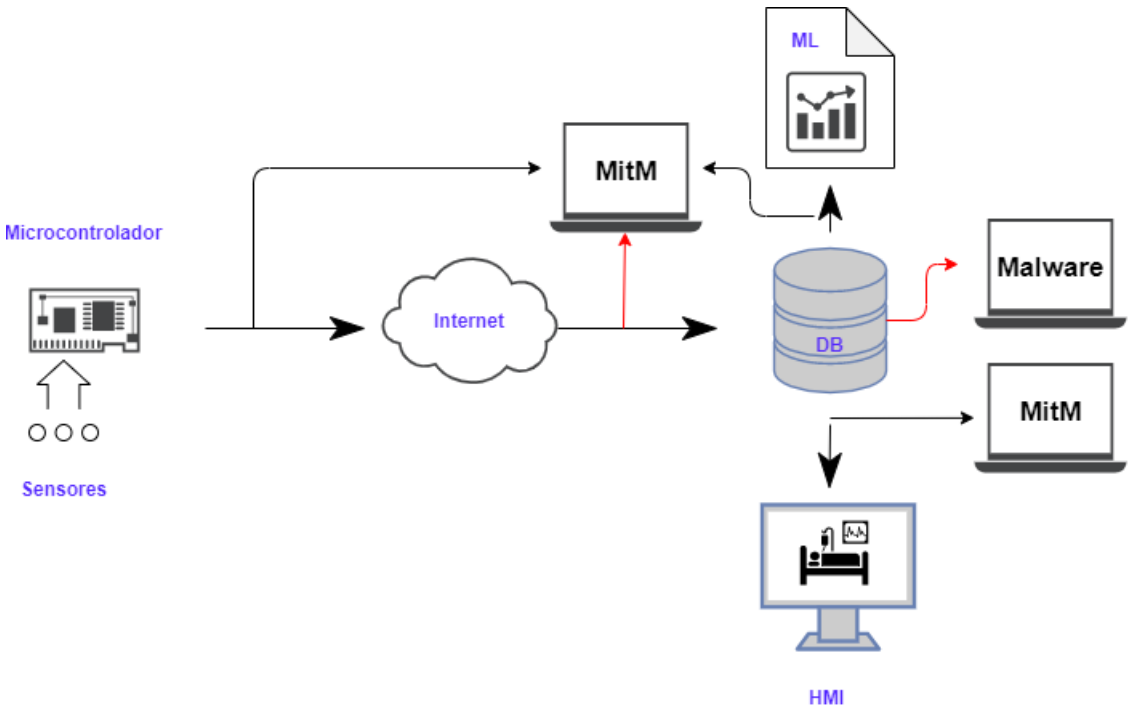


Figura 13: Esquema de puntos susceptibles

En la figura 13, que es una modificación de la figura 9, se puede ver que los ataques posibles del sistema se encuentran en las comunicaciones y en el propio equipo donde está instalada la base de datos. Al tratarse de un trabajo grupal y haber tenido que trabajar de manera remota por la situación de la COVID-19, los dos únicos puntos que se han podido atacar han sido los marcados en rojo en la figura 13 y estos son:

- Capturar la comunicación mediante un ataque Man in the Middle (MitM) entre el envío de datos de sensores y la recepción de dichos datos en la base de datos.
- Instalación de código o software malicioso en el equipo donde se encuentra instalada la base de datos.

El sistema de detección de anomalías está constantemente recibiendo datos del entorno real a través de los diferentes sensores, por lo que los ataques informáticos externos no serán los únicos a los que esté sometido el sistema CPS. De manera que, tenemos dos grandes grupos donde se clasifican los atacantes: **infiltrados**, en adelante *insiders* y **externos** [13].

- Los atacantes **insiders** son aquellos que se encuentran dentro del sistema y tienen acceso o forman parte de él [11] y en 2019 originaron el 70-80 % de los ataques a sistemas CPS [13].
- Los atacantes **externos** son aquellos que se encuentran fuera del sistema y mediante alguna técnica que se menciona en la sección 6.2.2 consiguen acceder al sistema.

Una vez localizados los puntos susceptibles del sistema e identificados los atacantes, en la siguiente sección se van a comentar los tipos de ataques que engloban los atacantes insiders y externos.

6.2. Amenazas en sistemas CPS

6.2.1. Ataques insiders

Estos ataques no siempre tienen que ser intencionados, y pueden ser originados por un error humano, a esto se le conoce como **participantes involuntarios** [17].

En el caso de los ataques **intencionales** que pueden ocurrir en el sistema tratado en este TFG son:

- Cambios en el sistema de climatización de la sala quirúrgica.
- Cambios en la presión de la sala quirúrgica.
- Movimientos en la mesa de operaciones.
- Instalación de malware o software malintencionado en el ordenador de la sala quirúrgica.
- Daños físicos en los diferentes sensores.
- Revelación de los modelos de predicción para realizar ataques de ingeniería inversa.

En cuanto a los ataques **involuntarios** tenemos los mismos que en el caso anterior a excepción de la revelación de los modelos de predicción. La diferen-

cia entre ambos ataques (intencionales e involuntarios) es la intencionalidad del mismo.

Exceptuando la revelación de los modelos de predicción y la instalación de malware en el ordenador de la sala quirúrgica, el resto de ataques afectan de manera directa a los datos que envían los sensores, por lo que la forma de detectarlos es a través de la detección de anomalías que se hace en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning.

En este TFG se incluye un script en Python, el cual actuaría como malware instalado en la máquina que aloja la base de datos MongoDB. Este script, haciendo uso de la librería Scapy, cada vez que se recibe una traza dirigida a MongoDB, selecciona los campos necesarios y se insertan en una nueva base de datos maliciosa. Ya que los datos no son redireccionados, si no, duplicados, la base de datos de origen sigue recibiendo datos y no sospecharía nada. Por lo tanto, una forma de evitar este ataque sería restringir de la mayor forma posible el uso de los equipos informáticos mediante la identificación del personal con usuario y contraseña.

En cuanto a la revelación de los modelos de predicción, la forma de detectarlo sería mediante la relación empresa-empleado, aplicando las técnicas que se mencionan en la referencia [18].

6.2.2. Ataques externos

En relación con los ataques externos, se pueden englobar en las siguientes categorías: ataques de revelación, ataques de engaño y ataques de interrup-

ción [13].

Estos ataques externos se producen en las comunicaciones. Las comunicaciones no cableadas de este sistema siempre son con la base de datos MongoDB, y esta nos ofrece varias soluciones en lo referente a estos ataques como se verán en la sección 6.3.

Ataques de revelación o divulgación:

Los ataques de revelación tienen el objetivo de hacerse con información útil del sistema. Esta información podrá ser usada por si misma o utilizada para elaborar ataques más complejos. Uno de los problemas de este tipo de ataques es el largo tiempo que puede pasar hasta ser detectados [13]. Por ejemplo, en el caso de este TFG, si se consiguen los datos de los sensores, podrían elaborarse una réplica de los modelos de predicción usados para detectar las anomalías en el TFG de Marta Ferrer Cuesta, Detección de anomalías en sistemas CPS mediante machine learning.

Ataques de engaño:

Los ataques de engaño buscan la forma de inyectar datos falsos o modificar los datos verdaderos del sistema. Concretamente, en el sistema de este TFG los datos que podrían ser cambiados son:

- Datos de los sensores a la base de datos.
- Datos de la base de datos a los algoritmos de machine-learning.
- Datos de la base de datos a la simulación.

Ataques de interrupción:

Los ataques de interrupción tienen la finalidad de dejar sin comunicación total o parcial a dos o más partes del sistema. MongoDB ante irregularidades en la recepción de trazas termina la comunicación para evitar ataques.

Los ataques ahora mismo mencionados no suelen ser realizados de manera aislada, sino que se complementan entre ellos. Teniendo esto cuenta, los ataques realizados en este TFG han sido:

Man in the Middle (MitM):

Mediante una máquina con sistema operativo Linux conectada a la misma red que la base de datos MongoDB, se ha interceptado la comunicación entre MongoDB y la máquina emisora de los datos de los sensores.

Para llevar a cabo este ataque se ha realizado un envenenamiento ARP de la red [10], suplantando las identidades del equipo donde se encuentra MongoDB y de la puerta de enlace.

```
arpsoof -i wlp7s0 -t 192.168.1.58 -r 192.168.1.1
arpsoof -i wlp7s0 -t 192.168.1.1 -r 192.168.1.58
```

Previamente se ha tenido que habilitar la redirección de paquetes para que las víctimas puedan seguir comunicándose.

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Una vez hecho esto, con algún *sniffer* de tráfico, como puede ser Wireshark, se pueden observar las comunicaciones entre el equipo donde se encuentra MongoDB y la puerta de enlace.

Modificación de datos de sensores MitM

Para la modificación de datos mediante un ataque Man in the Middle se ha desactivado el redireccionamiento de paquetes, y este envío se ha realizado de forma manual mediante Scapy.

```
echo 0 > /proc/sys/net/ipv4/ip_forward
```

Esto se ha hecho para que a MongoDB no le lleguen mensajes duplicados (original y modificación). Cuando se detecta que el paquete es de inserción, algún campo es modificado con el correspondiente envío del nuevo paquete.

En el campo de datos de los paquetes TCP que recibe MongoDB se encuentran mensajes en formato MongoDB Wire Protocol [27], por lo tanto, conociendo que posición ocupa el indicador de que ese paquete es un *insert*, se pueden filtrar dichos paquetes.

Al Scapy no tener implementado de forma nativa MongoDB Wire Protocol, ha sido necesario trabajar de forma manual con el campo de datos del paquete TCP en formato hexadecimal.

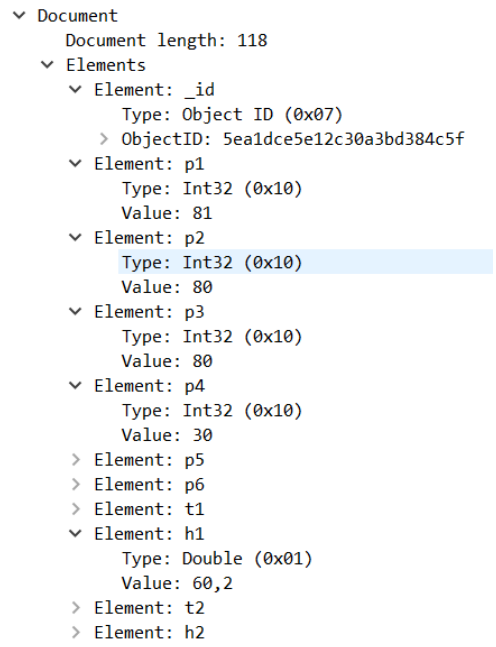
6.3. Seguridad en las comunicaciones

Para proteger las comunicaciones, MongoDB ofrece soluciones que se pueden definir a través de su archivo de configuración mongod.cfg.

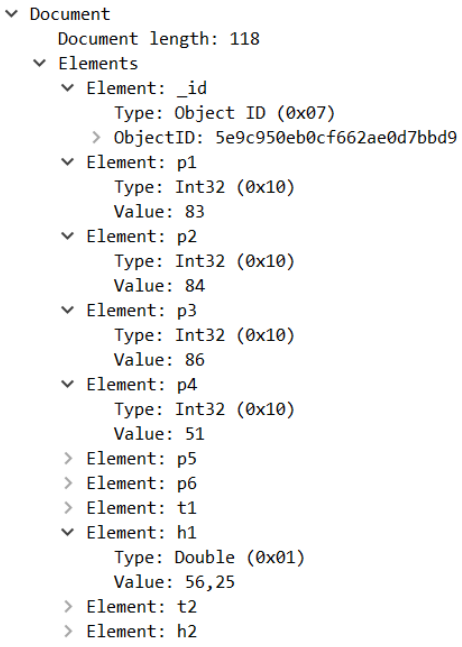
Uno de los primeros problemas que se encuentra en cuanto a las comunicaciones es el acceso a la base de datos por parte de equipos cuyas IPs no están registradas, por ello, MongoDB nos permite desde su archivo de configuración definir las IPs que pueden tener acceso.

```
net:
    port: 27017
    bindIP: 83.47.160.94
```

Aunque se restrinja el acceso por IP, por defecto, las comunicaciones con MongoDB están en claro, es decir, es posible leer el contenido de las trazas y esto dejaría al sistema susceptible de ataques de revelación. Con algún sniffer de tráfico que tenga implementado MongoDB Wire Protocol, será posible leer los mensajes. En la figura 14, se muestran dos trazas obtenidas con Wireshark que contienen un envío de los datos de los sensores.



(a) Comunicación en claro 1



(b) Comunicación en claro 2

Figura 14: Comunicación en claro

Para solucionar esto, MongoDB nos permite cifrar el canal de comunica-

ciones mediante TLS (Transport Layer Security) usando un certificado SSL [24]. Para este TFG se ha usado un certificado autofirmado, aunque esto implique no poder validar la identidad del servidor.

```
net:
  tls:
    mode: requireTLS
    certificateKeyFile: C:\Users\Certificates\mongo.pem
```

En este caso, cuando es capturado el tráfico, al ir cifrado el campo de datos de los paquetes TCP, Wireshark no es capaz de identificar dónde se encuentran las trazas de MongoDB Wire Protocol.

Siguiendo con los certificados SSL, MongoDB también nos permite validar la identidad del cliente. Esto lo hace comparando el certificado que le entrega el cliente con el indicado en el fichero de configuración. Además, para añadir más seguridad, podemos no permitir las conexiones que no presenten el certificado indicado (en este caso arduino.pem).

```
net:
  tls:
    mode: requireTLS
    CAFile: C:\Users\Certificates\arduino.pem
    allowConnectionsWithoutCertificates: true
```

Por último, para asegurar que nadie acceda a las comunicaciones sin autorización, se ha habilitado la autenticación con un usuario y contraseña.

```
security:
  authorization: enabled
```

Los usuarios que se han creado para este TFG son los que se pueden ver en la tabla 5 y como se índica se ha seguido el principio de mínimos privilegios [4]. Esto significa que a cada usuario se le han concedido los privilegios únicos que necesita para realizar sus funciones. Si en algún momento se necesitara ampliar algún privilegio de un usuario, éste, debería tener un tiempo de vida igual al tiempo que necesite el usuario ese nuevo privilegio, así podría seguir garantizándose el principio de mínimos privilegios.

7. Conclusiones y trabajos futuros

En este capítulo se van a comentar los problemas encontrados durante el desarrollo de este TFG, las conclusiones y una serie de ideas para futuros trabajos en esta misma línea. Pero para resumir, en este TFG se han establecido las bases de un sistema de monitorización de una sala quirúrgica mediante una representación 3D y el uso de sensores, además de analizar la seguridad en los sistemas CPS.

7.1. Problemas encontrados

Durante el desarrollo de este TFG se han encontrado distintos problemas que han tenido que ser solventados. Al ser un TFG dónde existen un gran número de tecnologías y tener muchos subsistemas interconectados, el número de problemas ha sido relativamente grande.

Como se comenta en la sección 4.1.3, se adquirieron tres sensores de temperatura y humedad, pero ante la imposibilidad de conectar más de uno a una sola placa hubo que adquirir una segunda placa. No se pudo adquirir una tercera placa, por lo que solamente se usaron dos sensores de temperatura y humedad.

En un primer momento, también se pensó en monitorizar el pulso cardiaco del paciente presente en la intervención quirúrgica. Se adquirieron tres sensores de pulso a través de la plataforma Ebay pero al llegar resultaron ser sensores falsos y la información que se obtenían a través de ellos era errónea.

Figura 15: Sensor de pulso

El precio en la página oficial de este sensor resultó ser elevado para el presupuesto con el que se contaba para este TFG grupal por parte de los integrantes del mismo, los cuál hacía imposible adquirir tres de ellos.

En relación a todos los sensores, debido al presupuesto, estos no pudieron ser de la máxima calidad. Esto hizo que en ocasiones poder obtener medidas adecuadas de todos ellos fuera algo difícil.

En cuanto al modelado 3D en Blender y Unity, al nunca haber realizado este tipo de tareas, llevó más tiempo del previsto pues hubo que aprender a usarlas desde cero.

La comunicación de Unity con MongoDB también llevó mayor tiempo del previsto, ya que fue bastante difícil encontrar los drivers de MongoDB que fueran compatibles con la versión de Unity.

Al usar la librería Scapy de Python para capturar el tráfico de la red, esta no tenía implementado el protocolo de MongoDB, MongoDB Wire Protocol. Ha sido necesario obtener los distintos campos de estos paquetes de forma manual para así poder trabajar con este protocolo.

Por último, la situación actual de la COVID-19 y el confinamiento dificultó el hecho de tener una base de datos donde los dos integrantes del TFG pudieran acceder desde distintas redes. Las placas Arduino y los sensores no

se encontraban en la misma red que la base de datos, pero finalmente se pudo solventar como se indica en la sección 5.2.1. Además, la hora de los ataques, el hecho de estar en diferentes redes también hizo que fuera más difícil realizar esta tarea.

7.2. Conclusiones

En este TFG se han sentado las bases de un sistema para la monitorización y alertas de anomalías durante una intervención quirúrgica. Además, se han localizado los puntos susceptibles de ataque en un sistema como éste.

Se ha trabajado sobre diversas tecnologías, lo cuál ha hecho tener que investigar constantemente sobre como poder implementarlas y comunicarlas. Desde un punto de vista personal, la realización de este TFG ha sido todo un reto personal por tener que aprender desde cero el mundo del modelado 3D y del hardware con Arduino, además de no haber trabajado nunca con una base de datos no relacional. Además, al ser un trabajo grupal, ha habido que tratarlo como si de un proyecto real se tratase, ya que ha sido necesario estar coordinados entre los integrantes del grupo y conocer siempre el trabajo de los demás.

También, se ha tenido la suerte de poder entrevistar a un cliente real y tener una visita a los quirófanos del Hospital de Alta Resolución de Benalmádena. Esto ha hecho poder tener información de primera mano sobre los procesos que se llevan a cabo en un quirófano y que se podía aportar por parte de los integrantes de este TFG.

Por último, la realización de este TFG me ha aportado un gran cono-

cimiento no solo en los aspectos más técnicos de una ingeniería, sino, en aspectos organizativos y de gestión a más alto nivel como pueden ser en este caso concreto gestionar un presupuesto de materiales o tratar con un cliente. Pero el mayor aprendizaje obtenido ha sido el poder solventar los numerosos problemas que han ido ocurriendo en el transcurso de este TFG.

7.3. Trabajos futuros

Al realizar este TFG se han pensado en que fuera escalable y en las posibles mejores que les podría realizar. Estas mejoras podrían ser:

- Visualización de información a través de gráficas sobre las anomalías que van ocurriendo en el sistema. Esto, acompañado de una interfaz gráfica la cual hiciera amigable todo este proceso.
- Inclusión de más tipos de sensores para poder monitorizar por completo la sala quirúrgica como puede ser un sensor de presión. Además, se pueden incluir sensores de monitorización de constantes vitales del paciente presente en la intervención quirúrgica.
- Sustituir las placas usadas por placas Arduino que tengan WiFi integrado o añadir modulos WiFi para no depender de un archivo Python para enviar los datos de los sensores.
- Para dotar de más seguridad a la base de datos, podría realizarse un TFG el cual solamente estuviera centrado en la seguridad de la base de datos usada en este sistema.
- Añadir una simulación de una intervención quirúrgica con animaciones

en Unity o un sistema de entrenamiento con la función de explorar escenario.

Referencias

[1] Hippa administrative simplification. Technical report, Department of Health Human Services USA, Marzo 2013.

[2] Smart hospitals: security and resilence for smart health service and infraestructures. Technical report, European Union Agency For Network And Information Security, Noviembre 2016.

[3] Marco para la mejora de la seguridad cibernética en infraestructuras críticas. Technical report, National Institute of Standards and Technology, Abril 2018.

[4] CISA Cibersecurity Infraestructure Security Agency. Least privilege. Accedido el 08/06/2020, https://www.us-cert.gov.

[5] Mohiuddin Ahmed and Abu S. S. M. Barkat Ullah. False data injection attacks in healthcare. In Yee Ling Boo, David Stirling, Lianhua Chi, Lin Liu, Kok-Leong Ong, and Graham Williams, editors, *Data Mining*, pages 192–202, Singapore, 2018. Springer Singapore.

[6] Descubre Arduino. ¿qué es el lenguaje arduino? Accedido el 13/08/2020, https://descubrearduino.com/lenguaje-arduino/.

[7] Atlassian. Trello. Accedido el 22/04/2020, `https://trello.com/es`.

[8] Atlassian. ¿qué es un tablero de kanban? Accedido el 22/04/2020, `https://www.atlassian.com/es/agile/kanban/boards`.

[9] Blender. Blender. Accedido el 22/05/2020, `https://www.blender.org/`.

[10] Instituto Nacional de Ciberseguridad. Arp spoofing. Accedido el 09/06/2020, https://www.incibe-cert.es/blog/arp-spoofing.

[11] Instituto Nacional de Ciberseguridad. Insider, las dos caras del empleado. Accedido el 30/06/2020, https://www.incibe-cert.es/blog/insider-las-dos-caras-del-empleado.

[12] Deloitte. ¿qué es la industria 4.0? Accedido el 21/07/2020, https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html.

[13] Seyed Mehran Dibaji, Mohammad Pirani, David Bezalel Flamholz, Anuradha M. Annaswamy, Karl Henrik Johansson, and Aranya Chakrabortty. A systems and control perspective of cps security. *Annual Reviews in Control*, 47:394 – 411, 2019.

[14] Elegoo. Elegoo hc-sr04 ultrasonic distance. Accedido el 20/05/2020, `https://www.elegoo.com/product/elegoo-hc-sr04-ultrasonic-distance-sensor-kits/`.

[15] Enrique Figueredo. Los hackers centran sus ataques en las redes hospitalarias en mitad de la crisis por el coronavirus. Accedido el 31/07/2020, https://www.lavanguardia.com/tecnologia/20200325/4889122708/hackers-coronavirus-hospitales-ataques.html.

[16] National Science Foundation. Cyber-physical systems (cps). Accedido el 21/07/2020, https://www.nsf.gov/pubs/2014/nsf14542/nsf14542.htm.

[17] LISA Institute. La amenaza de los insiders: cómo detectar la amenaza interna. Accedido el 30/06/2020, https://www.lisainstitute.com/blogs/blog/insiders-amenaza-interna.

[18] LISA Institute. Lista de 21 medidas para detectar y prevenir insiders en tu organización. Accedido el 02/06/2020, lisainstitute.com/blogs/blog/medidas-para-detectar-y-prevenir-insiders.

[19] Pardeep Kumar and Hoon-Jae Lee. Security issues in healthcare applications using wireless medical sensor networks: A survey. *sensors*, 12(1):55–91, 2012.

[20] Luis Llamas. El bus i2c en arduino. Accedido el 20/05/2020, `https://www.luisllamas.es/arduino-i2c/`.

[21] Macrovector. Operation room equipment isometric composition vector image. Accedido el 05/03/2020, https://www.vectorstock.com/royalty-free-vector/operation-room-equipment-isometric-composition-vector-19200524.

[22] Macrovector. Operation room equipment isometric composition vector image. Accedido el 23/05/2020, https://www.vectorstock.com/royalty-free-vector/operation-room-equipment-isometric-composition-vector-19200524.

[23] We Are Marketing. Metodología scrum: qué es y cómo funciona. Accedido el 22/04/2020, `https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html`.

[24] MongoDB. Configure mongod and mongos for tls/ssl. Accedido el 15/08/2020, https://docs.mongodb.com/manual/tutorial/configure-ssl/.

[25] MongoDB. Mongodb. Accedido el 23/05/2020, `https://www.mongodb.com/es`.

[26] MongoDB. Mongodb compass. Accedido el 21/05/2020, https://www.mongodb.com/products/compass.

[27] MongoDB. Mongodb wire protocol. Accedido el 21/05/2020, "https://docs.mongodb.com/manual/reference/mongodb-wire-protocol/.

[28] S.A. Noemí Sobrino, VP Retail EcoBuilding Schneider Electric España. Internet de las cosas y seguridad en hospitales. Accedido el 31/07/2020, https://hospitecnia.com/tecnologia/iot-internet-de-las-cosas/iot-seguridad-hospitales/.

[29] PyMongo. Pymongo documentation. Accedido el 21/05/2020, https://pymongo.readthedocs.io/en/stable/.

[30] Python. Python. Accedido el 13/08/2020, https://www.python.org/.

[31] Scapy. Scapy documentation. Accedido el 21/05/2020, https://scapy.readthedocs.io/en/latest/.

[32] C Sharp. Documentación c sharp. Accedido el 13/08/2020, https://docs.microsoft.com/es-es/dotnet/csharp/.

[33] Seeed Studio. Grove - temperaturehumidity sensor (high-accuracy mini). Accedido el 20/05/2020, "https://www.seeedstudio.com/Grove-Temperature-Humidity-Sensor-High-Accuracy-Mini.html.

[34] Unity. Unity. Accedido el 22/05/2020, https://unity.com/es.

[35] Unity3D. Codificación en c en unity. Accedido el 13/08/2020, https://unity3d.com/es/learning-c-sharp-in-unity-for-beginners.

[36] Unity3D. Monobehavior. Accedido el 28/05/2020, https://docs.unity3d.com/ScriptReference/MonoBehaviour.html.

[37] Wireshark. Wireshark. Accedido el 21/05/2020, https://www.wireshark.org/.

Apéndice A - Entrevista al cirujano

Preguntas generales

- ¿Cómo es la distribución de una sala quirúrgica básica (mesa de operaciones, pantallas, ecg, luces, puertas, ventanas)? ¿Cuántas salidas necesita tener el quirófano? ¿Y ventanas?

  Los quirófanos suelen ser cuadrados y tener una entrada amplia para que entren camillas. Además, cuentan con otro acceso, al área de sucio, zona en a la que se lleva el instrumental empleado en la operación. El instrumental limpio entra listo para ser utilizado por la puerta principal y sale por la puerta de sucio, es lo que se conoce como circulación de limpio y de sucio.

  Camilla en posición central y lámpara centrada encima de la camilla. Torre de gases al lado de la pared con brazo articulado. Aparatos de anestesia. Mesa instrumental. Sería bueno tener mesa para sentarse y escribir. La sala tiene que tener fácil desmontaje para limpiar rápidamente.

- Durante una intervención, ¿qué condiciones ambientales deben estar controladas? (nivel de temperatura, nivel de humedad, luz, etc.)

  El quirófano debe contar con unas buenas condiciones de luminosidad. Presión por encima de 5 mínimo – entre 5 y 10. Presión el quirófano superior a la de fuera. El aire siempre sale nunca entra. Lo más importante: humedad, presión y temperatura.

  Todas estas condiciones vienen detalladas en el manual: Bloque quirúrgico.

- ¿Qué constantes vitales del paciente deben controlarse? En cuanto al pulso cardíaco, cómo se mide, medidas normales, medidas de alerta, etc.

  Pulso simetría, electrocardiograma..

- ¿Qué tipo de robots quirúrgicos se están utilizando actualmente? (Qué tipo de operaciones se pueden realizar con dichos robots)

  Todos los robots que se emplean actualmente en quirófano son robots esclavos, lo que quiere decir que no tienen autonomía. Son los cirujanos los que ordenan los movimientos que deben realizar. Algunos de los que se están empleando actualmente son: Robot: Zeus (menos completo que el DaVinci). Zeus fue uno de los originales. DaVinci se controloa con la frente y pedales. Zeus es un manipulador de camaras. DaVinci en civil, virgen del rocio y sanis. Cada repuesto 5K o 6K y mantenimiento elevado. DaVinci es robot esclavo.

Preguntas específicas

- ¿En qué tipo de intervenciones se suele usar (el robot previamente dicho)? De esas intervenciones, ¿cuál es la que considera menos compleja?

  Una intervención simple sería la extracción de un lipoma.Aunque actualmente no son muy utilizados debido al alto coste del mantenimiento y de los recambios de piezas.

- ¿Cuáles son las fases de dicha intervención en condiciones normales?

  Intervención sencilla: Paciente llega al transfer, entra al quirófano, identificación correcta del paciente. Se le identifica con la pulsera (Comprobar nombre y fecha nacimiento), alergias, si es de intubación difícil,

confirmar procedimientos (de que se va a operar ej: hernia, de que lado). Luego se monitoriza, electrocardiografía, después anestesia y se entuba una vez dormido. Se controla la respiración. A continuación, pintar con antiséptico la zona de la incisión. Se colocan paños quirúrgicos y se prepara la instrumentación. Se toma el bisturí eléctrico y luego ya se procede a la operación. Cuando se termina se despierta al paciente y se lleva a la sala de recuperación y allí se monitoriza al paciente igual que durante la operación.

- ¿Qué tipo de "anomalías"(condiciones no normales) pueden ocurrir en una sala quirúrgica y que pueden afectar a la intervención?

  La anomalía más común en un lipoma es que se baje la tensión. Esto se controla por la pulxiosimetría. Cuando se detecta, se para la operación y se incorpora el paciente. Además, con respecto a los factores ambientales, si se detectan descompensaciones en estos valores y no se pueden recuperar la intervención se detiene.

- Independientemente del Robot, cuando se operan a personas, qué tipo de tecnologías o sistemas informáticos se aplican para controlar estados de la infraestructura y del paciente? ¿o no existen?

  Se controla, la humedad, presión y temperatura de la sala. Si la temperatura sube o baja de los valores normales, al igual que ocurre con la presión y temperatura se envía un aviso al sistema central.

- ¿Qué protocolos de emergencia existen cuando ocurre alguna anomalía (alarmas sonoras, visuales, luces de emergencia, etc.)?

  Suelen ser alarmas sonoras. Y luces de emergencia cuando se va la luz. En los robots suele haber un botón rojo para bloquear el sistema y poder manipular.

- Tipo de seguridad o política de seguridad (a nivel de infraestructura/-sala de operaciones) se establece para proteger las salas de cirugía.

  Para poder tener acceso a las salas de cirugía hay que disponer de una acreditación. La diferente instrumentación que es usada durante la intervención quirúrgica es controlada mediante códigos QR por lo que se conoce que material ha entrado en una sala.

  Por otro lado, para el suministro de medicamentos a los pacientes, solo se puede tener acceso a ellos mediante identificación del personal sanitario.

- Tipo de seguridad o política de seguridad (a nivel de información) se establece para proteger el acceso a las salas.

  Los equipos informáticos que se encuentran en las salas solamente son accesibles mediante usuario y contraseña. Además, para poder llegar hasta la sala existen otras medidas de seguridad que hacen más complejo el acceso no autorizado a estos equipos.

- ¿Cómo se gestiona la información que se controla de los pacientes? ¿Existe un sistema centralizado que recopila dicha información?

  Sí, la información de los pacientes se encuentra centralizada

Apéndice B - Manual de instalación

Bloque Arduino

Para que el sistema funcione de forma correcto, primero será necesario instalar las placas y los sensores. Los esquemas a seguir son los siguientes:

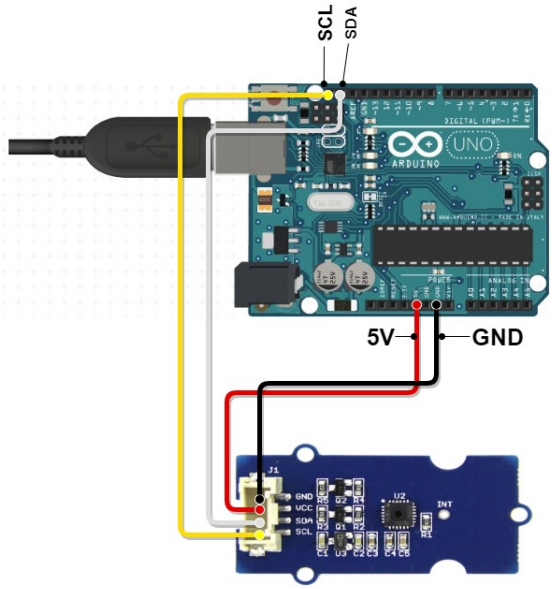


Figura 16: Esquema Arduino Uno

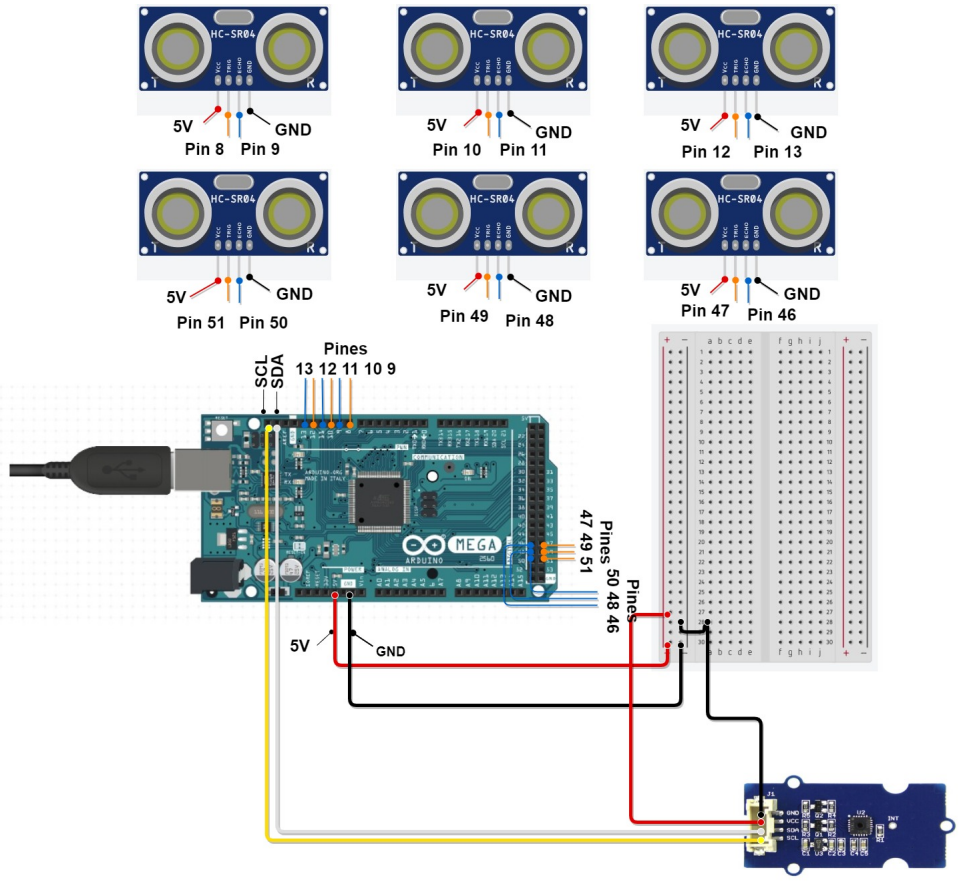


Figura 17: Esquema Arduino Mega

Para poder ejecutar los códigos de Arduino será necesario tener instalado el entorno de programación Arduino IDE [1].

Descargar los archivos con extensión .ino referentes a las placas Uno y Mega.

[1]Enlace de descarga de Arduino IDE: https: `//www.arduino.cc/en/main/software`

Bloque base de datos

Es necesario instalar una base de datos MongoDB [2]. Una vez instalada la base de datos habrá que crear 5 colecciones: packet_data_train, sensors_data_train, packet_data, sensors_data y predict_log. La instrucción para crear las colecciones es la siguiente:

```
db.createCollection("collection_name")
```

También será necesario crear los diferentes usuarios mencionados en la sección 5.2.2. Los usuarios se crean de la siguiente manera:

```
 db.createUser(
...
... {
... user: "user_database",
... pwd: passwordPrompt(),
... roles: [
... { role : "readWrite", db: "cps"}
... ]
... }
... )


* En roles incluir una lista con los roles requeridos
```

Antes de crear las colecciones y usuarios es necesario indicar la base de datos sobre la que se va a trabajar:

[2]Enlace de descarga de MongoDB: `https://www.mongodb.com/try/download/community`

```
use cps
```

También se pueden importar todas las colecciones mediante la siguiente instrucción:

```
mongorestore --Host = ip --port = 27017 -d directorio
```

Para cualquier consulta referente a MongoDB se puede consultar su manual [3].

Bloque Python

La versión de Python [4] sobre la que todos los códigos están escritos es la 3.6.10.

Las librerías necesarias para que los códigos se ejecuten de forma correcta son: Scapy, Pandas, TensorFlow, scikit-learn, Numpy, Keras y Pymongo.

Bloque Unity

Ejecutar archivo ejecutable (disponible para Windows 10, MacOS y Linux) o el instalador para Windows 10 desde el cual se ejecuta el escenario 3D.

[3]Manual de MongoDB: `https://docs.mongodb.com/manual/`

[4]Enlace de descarga de Python 3.6.10: `https://www.python.org/downloads/release/python-3610/`

Apéndice C - Manual de usuario

Una vez ejecutado el fichero de Unity proporcionado, se encuentra una pantalla como la de la figura 12. En ella encontraremos las opciones **Iniciar detección de anomalías** y **Explorar el escenario**. Las funciones de cada opción vienen explicadas en la sección 5.3.

Si se va a optar por ejecutar la opción Iniciar detección de anomalías, primero será necesario montar el circuito que ira conectado a las placas Arduino y asegurarse que dichas placas Arduino estén conectadas de forma correcta al equipo encargado de enviar los datos de los sensores. Los esquemas de los circuitos son los que se pueden ver en las figuras 16 y 17.

Seguidamente, será necesario compilar y enviar los programas arduino_uno.ino y arduino_mega.ino a las placas Arduino para comenzar la lectura de los datos de los sensores.

Antes de establecer la comunicación entre los sensores y la base de datos, será necesario iniciar la base de datos de MongoDB. Para ello, desde la ruta de instalación de MongoDB será necesario localizar la carpeta bin (Server\4.2\bin) y ejecutar el comando:

```
mongod --config mongod.cfg
```

Una vez la lectura de los datos de los sensores se este realizando de forma correcta y la base de datos este iniciada, es necesario establecer la comunicación entre ambos subsistemas. Se inician los ficheros Python send_measurements.py y packet_data_pred.py. Este último fichero almacena en la base de datos los datos de red de las comunicaciones con la base de datos.

- send_measurements.py: se inicia desde el equipo donde estén conectadas las placas Arduino.
- packet_data_pred.py: se inicia desde el equipo donde esté instalada la base de datos.

Antes de pulsar el botón Iniciar detección de anomalías es necesario iniciar los algoritmos de detección de anomalías (prediction.py, disponible en el TFG de Marta Ferrer Cuesta, Detecciióon de anomalías en sistemas CPS mediante machine-learning). Una vez hecho esto, ya se puede comenzar a visualizar las anomalías que ocurran.

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA